FUTURE DIRECTIONS IN
IN
DSS TECHNOLOGY*

MICHAEL TREACY

JANUARY 1985

CISR WP #123
SLOAN WP #1631-85

Center for Information Systems Research
Massachusetts Institute of Technology
Sloan School of Management
77 Massachusetts Avenue
Cambridge, Massachusetts, 02139

FUTURE DIRECTIONS IN
IN
DSS TECHNOLOGY*

MICHAEL TREACY

JANUARY 1985

CISR WP #123
SLOAN WP #1631-85

CENTER FOR INFORMATION SYSTEMS RESEARCH
SLOAN SCHOOL OF MANAGEMENT
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FUTURE DIRECTIONS IN DSS TECHNOLOGY *

by

Michael Treacy

Abstract

Decision support systems (DSS) technology is undergoing rapid innovation and evolution. DSS software facilities continue to become more comprehensive and may soon include a broad range of communications support features. We are also beginning to see that expert systems technology will enhance traditional DSS capabilities. The focus on hardware for DSS has shifted from the minicomputer and mainframe to the personal computer. Methods of physically integrating these devices are beginning to be applied, but logical integration has not yet begun. Key to logical integration is an overall software architecture designed to "glue" together diverse application packages at the data and interface levels. From discussions of these trends are drawn several implications for software vendors and for users of DSS technology.

*I wish to thank David De Long for his valued help on an earlier draft of this paper.
Funding for this research was provided by Citibank.

FUTURE DIRECTIONS IN DSS TECHNOLOGY

by

Michael Treacy

1 Introduction

The impact of all information technologies is in large part a result of the vision we hold for their use. Until recently, our view of the potential of decision support systems (DSS) has been restricted by the relatively limited capabilities of the technology and by our conception of DSS as an individual support system. Rapid advances in the technology, most notably in the area of microcomputers, and greater understanding of the potential impacts of DSS on the nature of managerial work now promise a change in the future role of DSS.

For the past fifteen years, decision support systems technology has evolved both steadily and predictably. Only recently have we seen rapid and revolutionary changes in DSS technology that have made obsolete many old concepts and assumptions about DSS capabilities. Understanding these recent changes and the future directions that they portend is one key to effectively managing the technology. Otherwise, one may continue to invest in an older generation of DSS capabilities that has significant limitations for leveraging a firm´s productivity.

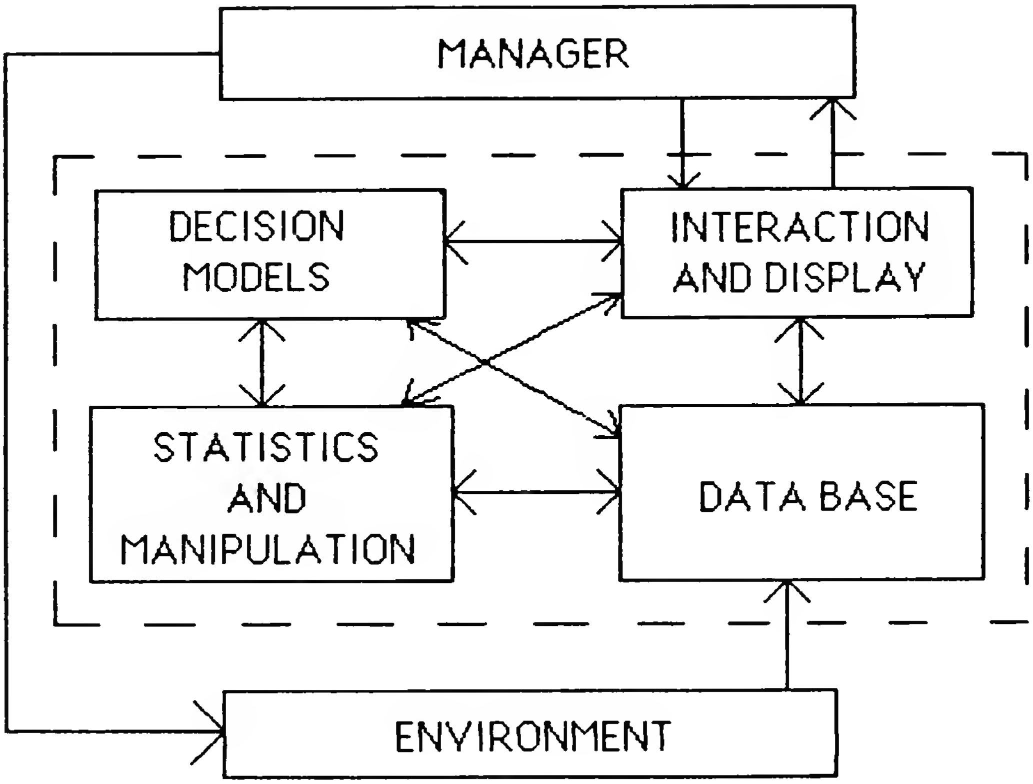
It is difficult to talk about decision support systems without acknowledging the definitional quagmire in which the term "DSS" exists. The fact that the definition of "decision support systems" remains problematic fifteen years after the term was introduced byGorry and Scott Morton [1971] is symptomatic of several problems. First, this is still a relatively new area, and both researchers and practitioners are trying to understand what the DSS concept represents in terms of design alternatives and its impact on the organization. Second, the competitive nature of the technology marketplace encourages vendors to turn virtually any MIS concept into a marketing buzzword without concern for the confusion caused by its misuse. Finally, DSS is a multi-disciplinary area drawing on fields such as behavioral decision theory, computer science, and systems analysis. Each discipline brings its own perspective and biases to the debate over the definition of DSS.

For our purposes, we can choose a fairly simple definition of DSS. We define a decision support system as a computer-based system used to support the needs of managers for data and analysis. This broad definition focuses on the functional capabilities of a support system and gives us wide latitude to explore alternative technologies.

Today, the evolution of data and analytic support is being driven by changes in the technology. We will be able to support new uses of systems and achieve new impacts on the business only if technology evolves to enable new types of systems. Therefore, understanding where the technology is going is fundamental to any discussion about the future of DSS. Paradoxically, for building systems today with existing technology, other issues such as needs assessment and design processes are more important.

2 DSS Technology Today

During the past fifteen years, a host of decision support system software facilities have been developed that allow analysts and managers to develop and use directly decision support systems. Figure 1, adapted from Montgomery and Urban [1969], shows the capabilities that have been provided by different DSS generators.

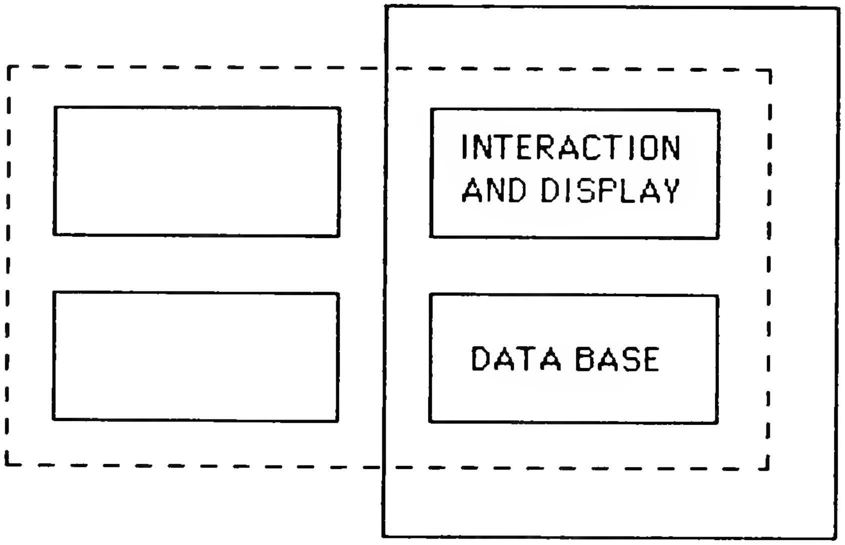


The Montgomery-Urban Model of DSS Capabilities

Figure 1

A decision support system provides a manager with another source of information on his or her internal and external business environment. Through an interaction and display facility that may include a command and data query language, report writing, and color graphics facilities, the manager can access a base of data, perform statistical, arithmetic, and other data manipulation functions, and create explicit models of his firm, his competitors, and the industry and economy.

Figure 1 represents an ideal set of DSS generator capabilities. In practice, the majority of software is of one of two types that fall short of the ideal. Figures 2 and 3, adapted from Wurts [1981], indicate these groups.



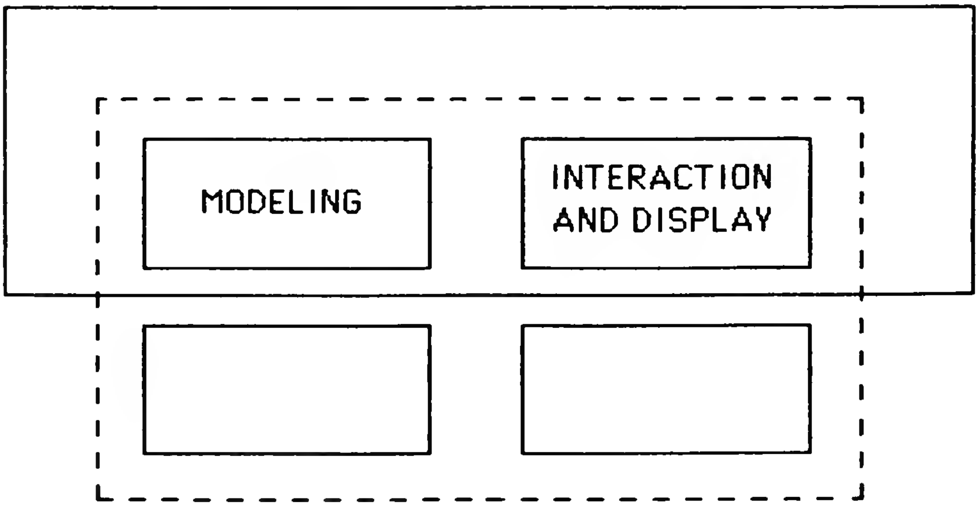
Friendly Data Base Management Systems

Figure 2

The friendly, easy to learn and use, data base management systems (DBMS) provide a manager with a facility for managing and accessing a large base of data, creating reports and graphs, and performing very limited analyses upon the data. They give managers the ability to choose the data that they wish to see and to format it in reports and graphs as they wish to see it.

The major weakness of friendly DBMS is indicated in Figure 2 by what capabilities they do not have. They do not provide adequate analysis or explicit modeling facilities for managers who have mastered data retrieval and are looking to manipulate data for analysis. In short, query systems do not provide a growth path for the typical manager who learns through a DSS to perform increasingly sophisticated analyses.

Figure 3 indicates the other major class of DSS generator. A spreadsheet modeling system gives a manager the ability to define an explicit model of several interrelated variables and to calculate the results of the model over several time periods. The packages usually allow a manager to define and solve a model, perform sensitivity and risk analysis on it, and generate reports and graphs. They are particularly well tuned to financial modeling, but provide no support for traditional mathematical programming models. They do not, in general, manage a data base or offer ad hoc analytic capabilities.

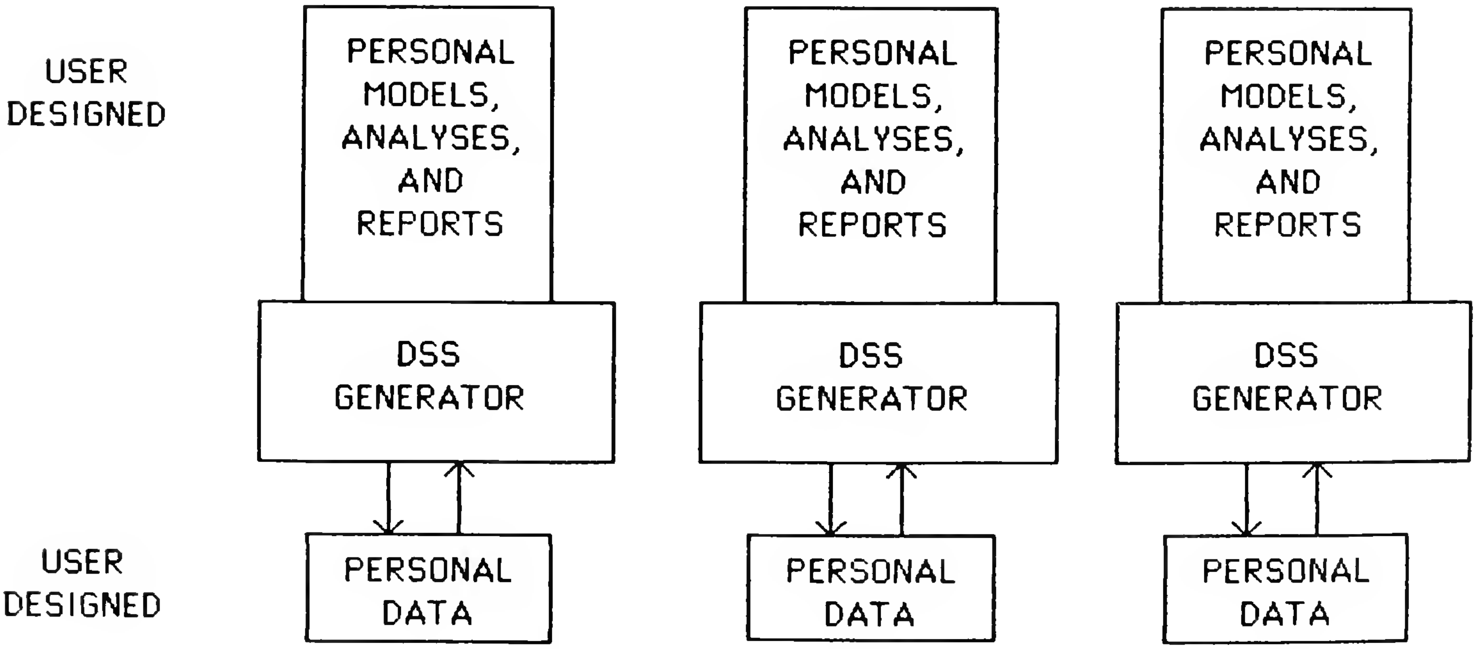


Spreadsheet Modeling Systems

Figure 3

Data base systems and modeling systems were first developed for large, shared computers about fifteen years ago. In the last few years the locus for new developments has shifted to the microcomputer. Important improvements have been made in the ease of use of spreadsheet modeling systems on microcomputers and this in part accounts for the extraordinary growth of micro-based DSS activity. Equivalent advances have not yet been made in micro-based data base packages and these have remained relatively less successful.

When modeling systems are used extensively in an organization, whether on a shared system or a micro, their use tends to look something like Figure 4. Each user of the system acts quite independently, with his or her own data, reports, and models that are separately maintained from the rest. There usually isn´t a data base management system

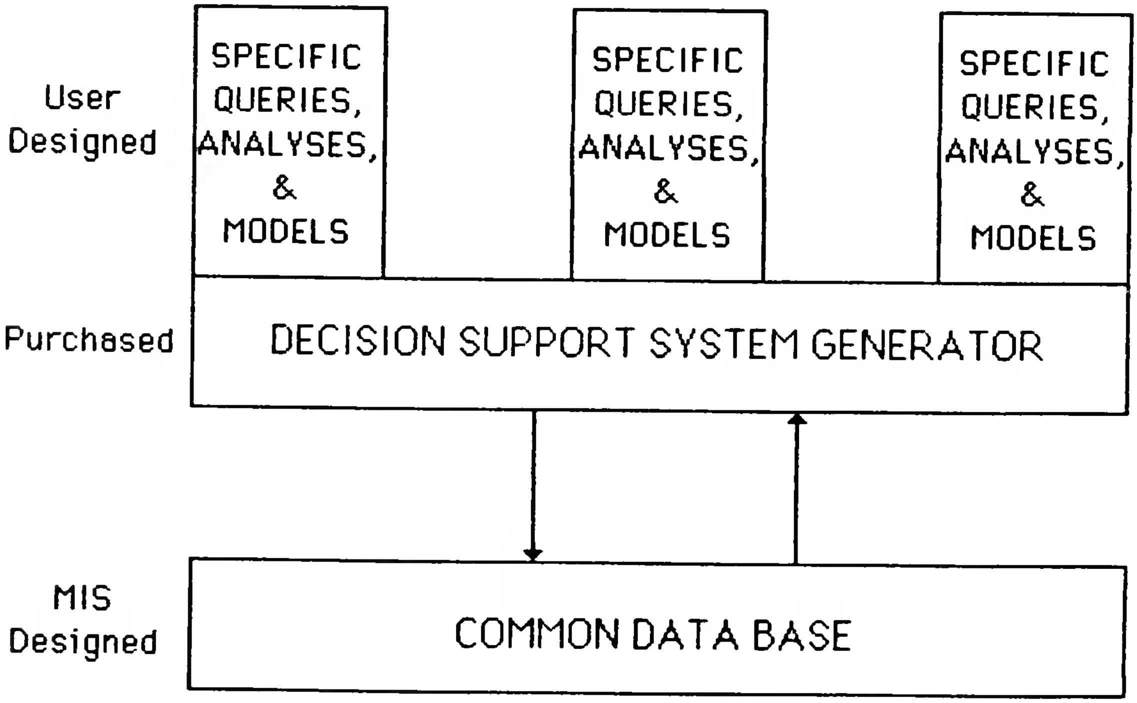

USER DESIGNED
PERSONAL MODELS, ANALYSES, AND REPORTS
PERSONAL MODELS, ANALYSES, AND REPORTS
PERSONAL MODELS, ANALYSES, AND REPORTS
DSS GENERATOR
DSS GENERATOR
DSS GENERATOR
USER DESIGNED
PERSONAL DATA
PERSONAL DATA
PERSONAL DATA

Individual DSS Developments

Figure 4

integrated with the modeling package, so data cannot be managed in a common, accessible pool.

Some companies have tried to create a degree of data and software commonality in their decision support systems environment. Instead of individual DSS development as shown in Figure 4, they have an organizational support system [Huber, 1982] that resembles Figure 5. This diagram illustrates three important features. First, users have a common decision support system generator. This facilitates sharing of models and analyses, reduces training needs, and helps create a mutually supportive environment among users. Second, users have a common base of data, designed and maintained with the support of information systems



Organizational DSS Developments

Figure 5

professionals. This facilitates sharing of data, reduces problems of redundancy and inaccuracy, and establishes elements of data resource management.

Finally, this scheme places great demands upon the DSS generator, for it must be all things to all people. It must manage data, have easy to use retrieval facilities, and powerful modeling and analysis capabilities. It must combine all four capabilities shown in Figure 1, and in a fashion that makes it appealing to a diverse range of users, with often narrow interests or needs. A very limited number of packages have come anywhere close to meeting these criteria and these packages have all been mainframe/mini based, where sharing of data is relatively easier. The present challenge is to provide a common and integrated DSS software architecture across a distributed hardware environment.

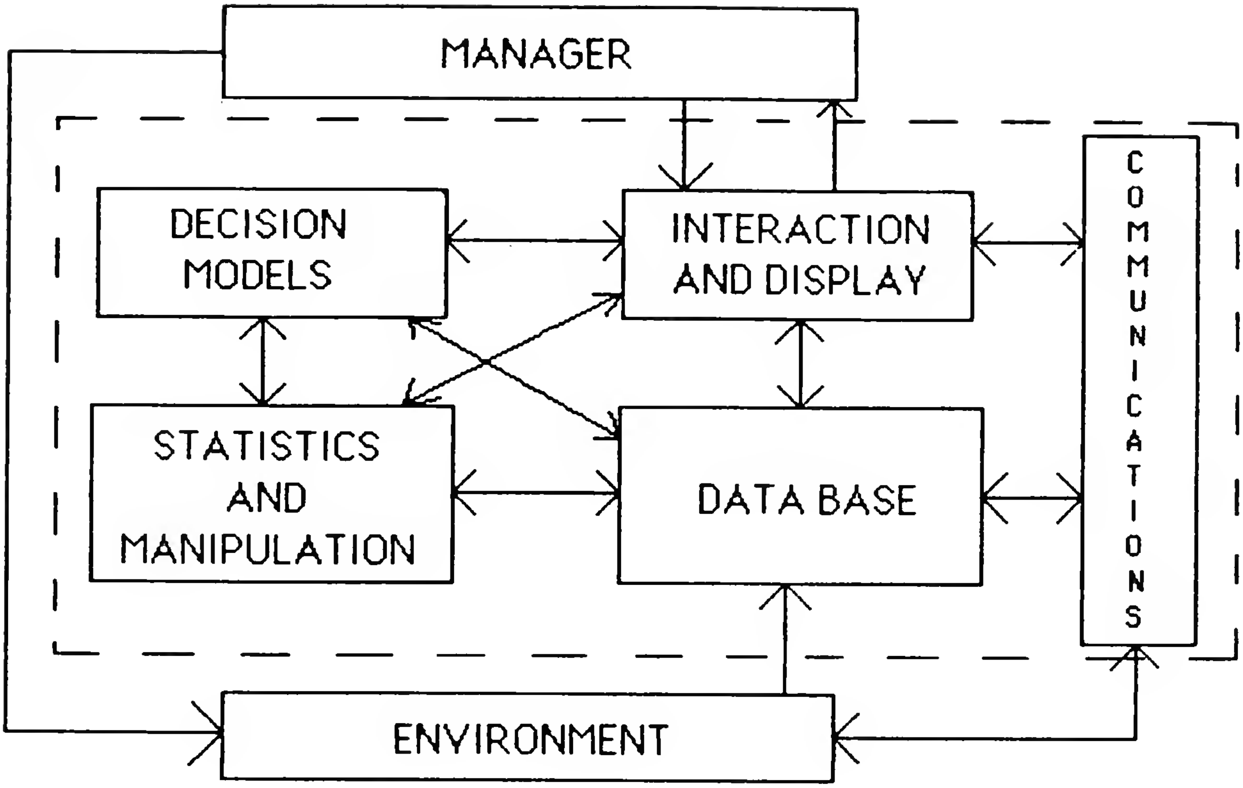
3 Changes in Software Functionality

In the area of software capabilities, there are three major trends. First, existing packages are becoming more comprehensive in terms of the four basic capabilities of DSS software: modeling, *ad hoc* analysis, data base management and interfaces. Packages that previously only had modeling and interface capabilities are adding data management and *ad hoc* analysis capabilities. Meanwhile, easy-to-use data base management packages are improving their interfaces and adding modeling features. In short, DSS software packages are growing up to fill all of the four basic DSS capabilities. "Integrated" has become the catchword of the DSS software industry.

A second advance that will have a major impact on how we view and use decision support systems will be in the area of electronic messaging. Advanced technology in this area is just becoming available. It promises to go well beyond usual notions of computer-based communications, such as electronic mail, by providing image, voice and possibly video in addition to alphanumeric communications.[Sirbu, 1978] The personal computer is evolving to become an extended telephone, offering a manager the services of telephone, computer, electronic mail, facsimile, photocopier, and television in a single device. Messages that use several media of communications will be composed and sent with the one system.

The relevance of this technology to decision support systems is that it represents a shift from early concepts of DSS that made no allowances for communications. The original idea of DSS, as represented in the Montgomery-Urban model did not explicitly consider the individual in an organizational context. Given this new electronic messaging technology, however, there is another capability of support systems that will grow up, communications support. Figure 6 includes the role of communications capabilities in the Montgomery and Urban framework.

Whether we call communications a subset of DSS or an allied capability is a semantic debate. Regardless, electronic messaging will be a major new technological capability that will impact how we view needs for support, how we design and implement systems, what impacts they have, and what policies we need to have in place to manage their development.



A Modified Model of DSS Capabilities

Figure 6

The third trend, one that is only beginning, is toward the use of expert systems technology to improve each of the four basic capabilities of DSS software. Expert systems is a branch of artificial intelligence that is concerned with building computer systems that display expert reasoning abilities.[Davis, 1982; Winston, 1984; Hayes-Roth, Waterman, and Lenat, 1983] The DSS field is going to borrow from the toolkit that has been developed to build expert systems. For example, we will see much more intelligent interfaces, systems that try to understand what the user is attempting, or that will reshuffle menus, or provide help systems automatically, or ignore typing mistakes, or accept voice input. These interfaces will make support systems much easier to use.

In the data base management area, to date DSS software has only handled quantitative data. But there is a lot of knowledge that does not fit into quantitative formats, and expert systems technology provides a way for us to capture and use that qualitative knowledge. For example, sales call reporting systems contain useful bases of textual information about customer histories and trends. Yet, because that information is non-quantitative, analysis of that data is outside the present bounds of DSS. In the future, with the use of expert systems technology, we should expect to be able to call upon data in text format and utilize its semantic content for analytic purposes.

In addition, with expert systems technology *ad hoc* analysis tools will become smart tools. Imagine a system with forecasting capability that helps teach the user how to forecast. Or, the system might observe seasonality in a set of data and build it into the forecast, while explaining its actions to the user. Gale and Pregiban[1982] have built an expert regression system which "emulates some of the interaction between a client and an expert statistical consultant."[p. 110] Basic *pro forma* accounting models and standard accounting analyses will become a part of modeling systems. The knowledge represented by financial accounting interpretation can be built into software using expert systems tools.

Over the past twenty years, a branch of set theory known as fuzzy sets has developed and is beginning to influence our ideas about modeling systems.[Bellman and Zadeh, 1970] The opportunity is upon us to produce fuzzy modeling capabilities. These models will remove the traditional constraints that models must be precisely stated, that model rules must not conflict with each other, and that the logic of the model must be essentially complete. Fuzzy modeling will allow the user to externalize the sometimes fuzzy, incomplete and inconsistent thinking or rules that most managers have to grapple with when making decisions.

Fuzzy models are used by managers all the time. For example, in a pricing decision, a manager might bring into play the following rules:

(1) Our price should be about two times direct costs.

(2) Our price should be just below our dominant competitor´s price.

(3) If our competitors´ prices go too high, we should price for increased market share.

None of these statements is in a form precise enough to be used in a standard modeling system. Each statement begs for refinement. With a fuzzy modeling system, these statements form a model of the pricing decision that could be solved. The results of that solution clearly would not be satisfactory, but it is a useful starting point that prompts a manager for a more refined representation of his or her mental model.

Finally, communications capabilities can be significantly enhanced with expert systems technology. Effective communications comprises more than just the transportation of messages. Several communicaitons functions require the intelligent application of knowledge about messages. For example, in telephone and mail systems, the functions of filtering and categorizing are essential for effective communications.

Many of these functions can be incorporated into support systems using expert systems technology.

4 Changes in Hardware Environment

The evolution of DSS software functionality is complicated by the current major migration of software capabilities from the mainframe and minicomputer to the microcomputer.[Healey, 1983] This is a major shift, one that may well move all analysis systems off shared systems in the next three years. From a software architecture standpoint, there are sound reasons for this migration.

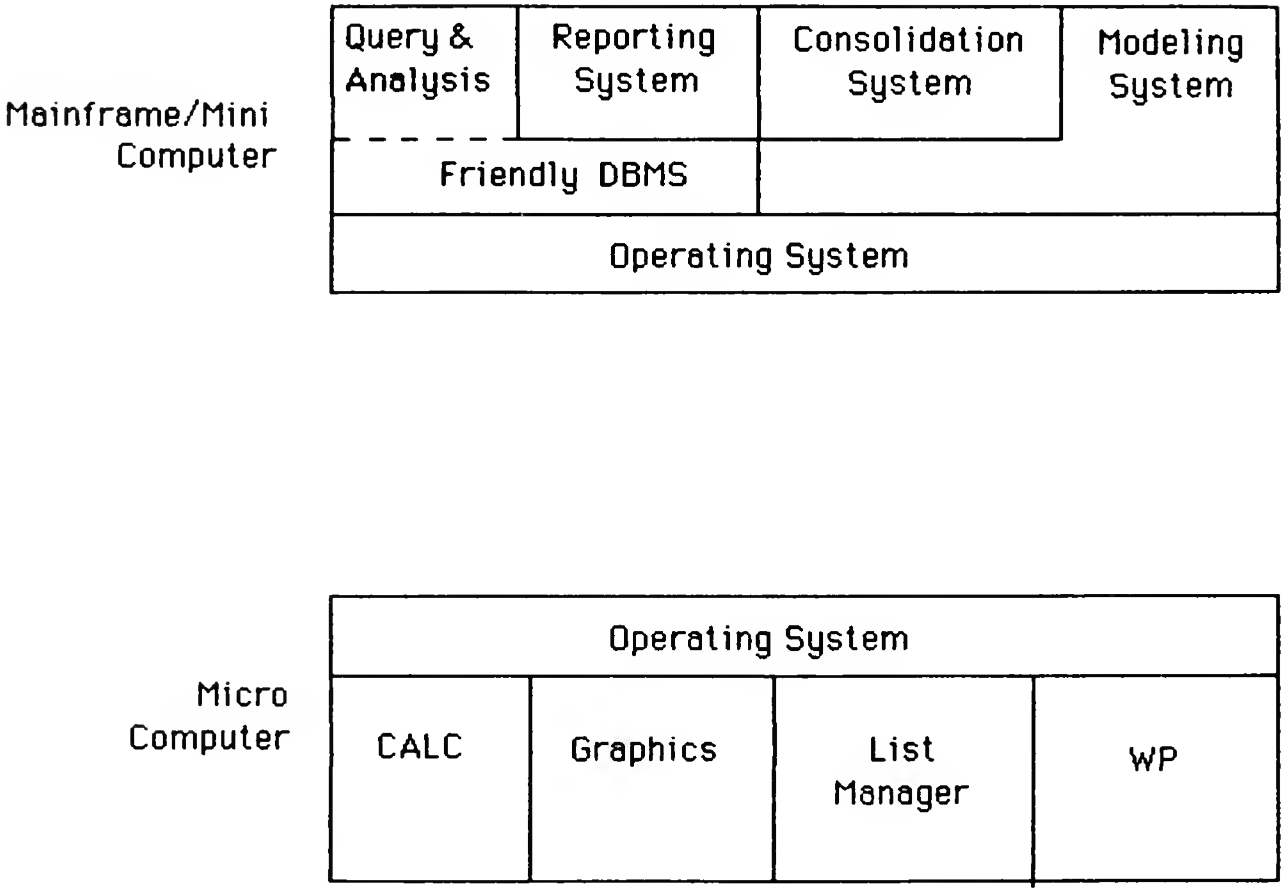
As demonstrated by the phenomenal success of micro spreadsheet packages, ease of use is one of the most important technical features of management oriented software packages. A key to ease of use is the degree of interactivity, for that determines the rate of feedback and responsiveness of the system.[Doherty and Kelisky, 1979] Microcomputers provide very high limits on interactivity. Any location on the display of a microcomputer can be changed virtually as quickly as a primary memory location can be changed. Thus, the entire screen can be transformed in the blink of an eye. On a mainframe or minicomputer, highly interactive, easy-to-use interfaces cannot be delivered to asynchronous terminals because of the limited bandwidth between the screen and the processor. Even using a 9.6 kilobit connection, it takes two seconds to send a full screen of characters. Graphics take even longer.

Eventually, this handicap will eliminate the mainframe and minicomputer from decision support systems, except as data repositories and device managers on a network. How quickly this occurs will depend on how quickly IBM introduces subsequent generations of microcomputers, which will, ultimately, be many times more powerful than its original PC. Their recent introduction of the PC AT is an very important step in this direction. The AT offers faster processing, much more memory, and expanded and faster secondary storage. It will enable an entirely new generation of decision support systems software to come to market within the next two years.

As a result, the only role left for the mainframe in the future of DSS will be that of a central data manager and as a manager of shared devices, such as printers and optical scanners. There is a serious question as to whether or not the software vendors now producing mainframe and minicomputer based DSS generators will be able to develop products for the new marketplace that is evolving.

5 Impacts of Hardware Changes upon Data Access

At present, the decision support systems hardware environment can be characterized as one of double innocence. Mainframe and mini based systems are almost completely separate from microcomputer systems, without sharing of data, software, or users. Figure 7 illustrates this. On the mainframe or minicomputer can be found friendly DBMS and modeling packages offering the usual run of capabilities: query, ad hoc analysis, and modeling. These software tools are often used as application generators to create reporting and consolidation systems that were once written in COBOL, but their main use is for decision support. Four major capabilities can be found on the microcomputer: spreadsheets (modeling), graphics, list management (a crude DBMS), and word processing.



Existing DSS Software Architecture

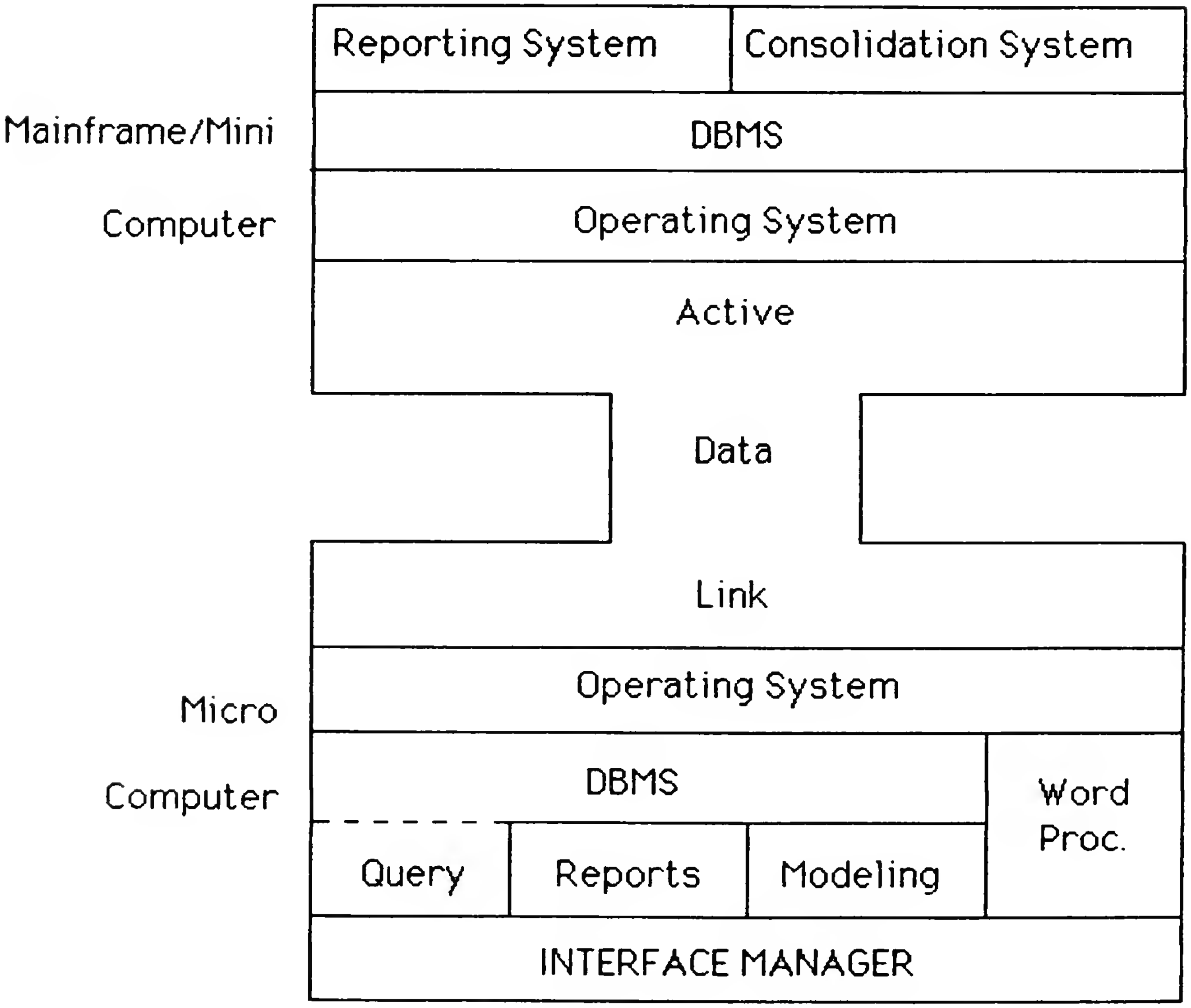
Figure 7

There are two strategies among mainframe DSS software vendors that we have seen as a defensive reaction to the migration to the micro. One is to create microcomputer versions of DSS packages that previously existed only on the mainframe.[Ferris, 1983] The problem with this strategy is that typically the new micro products are merely replicas of the mainframe packages that fail to take advantage of the unique capabilities that a microcomputer affords, particularly for creating highly interactive interfaces. Outside their existing base of customers, these products are often completely uncompetitive.

The second strategy is to offer the microcomputer user a tailored terminal emulation capability. This turns the microcomputer into a smart terminal for accessing the mainframe or minicomputer resident software. Smart features, such as command editing, capturing output to floppy disks, and menu and mouse oriented command building, can be added by using the local power of the microcomputer. Some of these terminal emulators also provide file formatting capabilities, so that the user can bring data down to a microcomputer in a format used by microcomputer software.

The terminal emulation strategy is really a stop gap measure that allows mainframe software vendors to quickly announce microcomputer products. It is not a long term solution. The major problem with this strategy is that the data link between mainframe or minicomputer and microcomputer is completely passive. A user who wishes to use data from a mainframe in his or her microcomputer´s modeling package must: (1) invoke the terminal emulation facility, (2) use the mainframe package´s command language to retrieve the appropriate data, (3) transfer the file in the correct format, (4) terminate the emulation facility, (5) invoke the modeling software, and (6) command the package to read in the transfered file. Preferable is to be able to call for the correct data while in the modeling package, using consistent commands.

A more sophisticated approach to linking micro-based DSS users to mainframe data is being taken by a few traditional DBMS vendors and is illustrated in Figure 8. The design involves three components: a mainframe or minicomputer based DBMS, a microcomputer based DBMS, and software to join them together, what we call an active data link. The active data link maintains a directory of data available to the user on the mainframe system. If a microcomputer application makes a call to



Emerging DSS Software Architecture

Figure 8

the micro DBMS for data that it has, the data is furnished. If data is requested that it does not have, the request is passed to the active data link, which checks the mainframe data directory, issues the appropriate procedure calls, receives the data from the mainframe, and passes it through the micro DBMS to the application. Thus, the mainframe DBMS appears as a virtual resource of the micro, an extension of the microcomputer´s own data base management system.[Goldstein, et al, 1984] The user has a unified view of both micro and mainframe resident data and does not need to be concerned with its location. To make this design work, the micro applications software for query, analysis, and modeling must be rewritten to run against the micro DBMS. For this reason, early products of this type have necessarily included a full suite of microcomputer applications.

Data base packages are needed on microcomputers not primarily as stand alone applications, but as systems software to enhance the integration of diverse applications software. For example, when a manager uses a set of applications packages such as graphics, spreadsheet, and word processing with DBMS capabilities behind them, data from one application remains available in the DBMS so that it can later be pulled into another application. In this case, the data base management capabilities are being used behind the applications packages to integrate them.

The evolution of DSS functionality to incorporate multimedia communications capabilities will necessitate the development of systems for managing non-quantitative data in an office environment. Designs for these are only beginning to emerge [Ahlsen, et al, 1984;

Zdonik, 1983] Incorporating these designs into a distributed scheme has also been studied.[Lyngbaek and McLeod, 1984] It may be several years before commercial systems are widely used.

A very important issue in this design is the development of standard interfaces between layers of software. Standards are crucial if a diversity of products are to work together. Standards can develop through cooperation of vendors and the scientific community or through the market power of a particular vendor.[Sirbu and Zwimpfer, 1984] The open systems interconnection data communictions standard developed by a subcommittee of the International Standards Organization is an example of the former.[Folts, 1981] Microsoft´s Disk Operating System (DOS) is an example of the latter.[Microsoft Corp., 1983] For this next generation of DSS software, the major commercial DBMS vendors are best positioned to invoke _de facto_ standards for data access.

6 The Evolution of Interface Design

Yet another change we will see in DSS software comes about because of the revolution taking place in interface design. Traditionally, mainframe systems have been programming oriented, offering virtually no help to the user who lacks a conceptual understanding of how the data is stored in the computer, or who doesn´t know how to respond to curt system prompts such as "enter." By following some fairly simple design rules, many of the more difficult features of traditional, programming oriented interfaces can be eliminated.[Branscomb and Thomas, 1984] But in the future we will see some very untraditional interfaces on data base software packages, much like those that are now standard on microcomputer-based modeling software, such as Lotus 1-2-3.

Unlike a standard command oriented DSS. interface, the orientation of microcomputer-based modeling packages is always to show the results the user is seeking, while the algorithm that shows how the results were derived is suppressed.[Kay, 1984] Unlike mainframe packages which traditionally have been programming oriented, the ease of use of such results oriented interfaces significantly reduces the barriers to use for potential DSS users.

In present data base query languages, the user must write the exact query for what is wanted if results are to be produced. No intermediate results are available, which makes it a process akin to target shooting. Only when the user aims correctly with the right query is the appropriate answer provided. Given the importance of data retrieval in decision support systems, existing software is frequently inadequate for the user´s needs. But a new generation of visual retrieval language will change that. It will open up ways of viewing data retrieval as a process of zooming in and out of a data base until the right data are found.

The emergence of interface managers as separate software packages will facilitate the development of better interfaces on application packages. Interface managers sit "in front" of applications and provide window management and other tools that application developers can use to create friendlier interfaces (see Figure 8). In the future, a user will interact "through" an interface manager with a microcomputer-based application that relies upon a DBMS to manage its data. This is a far more complex systems software world than exists today on a microcomputer, but it is one that major vendors are actively trying to build because it will facilitate the development of more powerful applications with less effort.

7 Implications for Software Vendors

This vision of DSS in the future was not obtained by gazing deeply and intently into a crystal ball. It was formed through discussions with product strategists in several dozen leading and emerging DSS software companies. The best of the vendors have the power to create the future of DSS technology. For the rest, this forecast defines the emerging dimensions of future competition.

Four distinct classes of software are beginning to emerge: (1) distributed data management software, (2) microcomputer interface managers tightly coupled to operating systems, (3) applications software on the microcomputer that works in conjunction with the interface manager and local DBMS, and (4) applications development software on the mainframe for creating reporting and consolidation systems.

Distributed data management is a difficult technical problem. Major offerings in this area are likely to come from vendors of traditional transaction-oriented DBMS. These firms have products that can be adapted for use in a distributed system, they can acquire rights to a microcomputer DBMS, and they have the technical talent to build an active data link. Vendors of friendly DBMS presently used for DSS are in a relatively poorer technical position for making the transition. Their added value has not been the sophistication of their data management, but the ease of use of their interface. With query languages moving down to the microcomputer in much more visual and interactive forms, much of this value will be lost. Many of their products have neither the capacity nor the sophistication to be used as a central DBMS accessed by hundreds of microcomputers. Significant investments will have to be made. Where these firms do have an important edge over traditional DBMS vendors is in marketing. This is a very important advantage that may make it difficult for traditional DBMS vendors with sophisticated products to compete in the end user marketplace.

The marketplace for interface managers coupled to operating systems is coming down to a two horse race. Microsoft, the owners of the Disk Operating System (DOS) have an announced product called Window. IBM, which is expected to make its move into operating system shortly, has announced an interface manager called Topview. Which horse will ultimately win this race is an easy bet.

Of great interest over the next few years will be the application vendors such as Lotus and Ashton-Tate. These firms have tremendous marketing power, but little "systems" experience, so it may be difficult for them to compete in distributed data management software, except through a strategic alliance with a DBMS vendor. Follow-on innovations have been hazardous in microcomputer software and there is still room for new vendors to sweep them aside. The IBM AT hardware is of such power that it will facilitate a new generation of applications software that may obsolete present applications.

Finally, a safe harbour for battered mainframe software vendors may be found by some as mainframe applications development facilities. Demand for these systems is growing rapidly, as data processing organizations discover that large productivity gains can be made.[Lientz and Swanson, 1980] Application such as reporting and consolidation systems sit somewhere between a transactional or operational system and an end user facility. They are generally high maintenance, run infrequently, so that machine efficiency is not important, and written by IS professionals. Comprehensive mainframe DSS software, with a little adjustment, provides an excellent development environment for these systems.

Only a relatively small number of firms will be able to compete successfully in more than one of these software segments. Not only is the technology different in different segments, but the customer is different as well. For distributed data base products, the primary prospects are the data resource manager and the DBMS technician. Applications software is sold to end users and their support professionals. Application development facilities are bought by application development staff within information systems. The development of accepted standards between classes of software will further the separation between these four submarkets.

8 Implications for Management Policies

Mainframe and minicomputer-based decision support systems software is nearing the end of its life cycle. Nonetheless, corporations continue to purchase this software, though in declining quantities. These packages are purchased to maintain compatibility with existing systems, to avoid the risks of unproven, newer technology, and because many of the microcomputer-based options are still evolving or not well known. In future years, these will rarely be satisfactory reasons for purchasing host-based DSS software.

The most common reason for continuing to build a mainframe or minicomputer based end user computing environment is that it preserves the value of the existing investments in older systems. This is true, but that value is maintained at the expense of even greater value that will be derived from a newer generation of microcomputer based DSS technology. What will emerge over the next few years is a set of software capabilities that will lift support systems to a new plateau of impact. That level cannot be bridged from the mainframe base of DSS technology.

This is not to say that much of the investment in existing DSS systems cannot be preserved. Instead it argues that preserving the existing investment should not be a goal or constraint in making future investments. A key observation is that newer technology will impact data management oriented and modeling oriented DSS software at different rates. Data management for shared data bases logically should continue to reside on the mainframe, so companies that have built mainframe or mini based data base oriented systems will be in a much better position to integrate the new generation of DSS capabilities into their existing systems, thereby preserving their investments. With the analytic portion of decision support systems software moving to the microcomputer there will be a nice marriage between the old and new technologies. On the other hand, organizations with large installed bases of mainframe and minicomputer based modeling packages have a big job of data base building ahead of them before they can take advantage of the new technologies. In this case, investments in early DSS will probably have to be written off.

Major errors in purchase decisions can be avoided if a firm´s end user computing policies include a plan for the evolution of its technological infrastructure.[Henderson and Treacy, 1984] The issue of technological infrastructure includes policies defining appropriate hardware, software, and communications equipment for the DSS environment. Any plan should include a forecast of technological developments and of the evolving needs of the organization. It should have a horizon of about five years and should set down a schedule for the phased introduction and assimilation of newer generations of technology as they come available.

Internal corporate standards are a key component of a technology plan for end user computing. Software standards promote the ability to share analyses and data and simplify support and training. But standards can also act as a barrier to newer, more innovative generations of technology. Standards can sensibly be used to phase the introduction of new technology. They should not be used to freeze it.

Successful new software products emerge as a response to pressures placed upon vendors by their competition and their customers. Most vendors strongly desire more help from their customers in shaping future product offerings, so that they can better meet the needs of the marketplace. Corporate customers that develop that dialogue with vendors can gain sharp insights into future directions in DSS technology. And that vision of the future, after all, is the basis for proactively managing corporate end user computing.

REFERENCES

Ahlsen, M., Bjornerstedt, A., Britts, S., Hulten, C., and Soderlund, L., "An Architecture for Object Management in OIS," ACM Transactions on Office Information Systems V. 2, N. 3 (July 1984), p. 173-196.

Bellman, R.E., and Zadeh, L.A., "Decision-Making in a Fuzzy Environment," Management Science, V. 17, N. 4 (Dec. 1970), p. B141-B164.

Branscomb, L.M. and Thomas, J.C., "Ease of use: A system design challenge," IBM Systems Journal, V. 23, N. 3 (1984), p. 224-235.

Davis, R., "Expert Systems: where are we and where do we go from here?" AI Magazine, Summer 1982.

Doherty, W.J. and Kelisky, R.P., "Managing VM/CMS systems for user effectiveness," IBM Systems Journal, V. 18, N. 1 (1979), p. 143-162.

Ferris, D., "The Micro-Mainframe Connection," Datamation, V. 29, N. 11 (Nov. 1983), p. 126-138.

Folts, H.C., "Coming of age: A long-awaited standard for heterogeneous nets," Data Communications, Jan. 1981, p. 63-73.

Gale, W.A., and Pregibon, D., "An Expert System for Regression Analysis," Computer Science and Statistics: Proceedings of the 14th Symposium on the Interface, 1982, p. 110-117.

Goldstein, B.C., Heller, A.R., Moss, F.H., and Wladawsky-Berger, I., "Directions in cooperative processing between workstations and hosts," IBM Systems Journal, V. 23, N. 3 (1984), p. 236-244.

Gorry, A. and Scott Morton, M., "A Framework for Management Information Systems," Sloan Management Review, V. 13 (Fall 1971), p. 55-70.

Hayes-Roth, F., Waterman, D., and Lenat, D., Building Expert Systems, Addison-Wesley, Reading, MA, 1983, 444 pp.

Healey, M., "Junking the Mainframe," Datamation, V. 29, N. 8 (Aug. 1983), p. 120-136.

Henderson, J.C. and Treacy, M.E., "The Management of End User Computing," CISR Working Paper #114, April 1984.

Huber, G.P., "Organizational Information Systems: Determinants of their Performance and Behavior," Management Science. V. 28, N. 2 (Feb. 1982), p. 138-155.

Kay, Alan, "Computer Software," Scientific American, V. 251, N. 3 (Sept. 1984), p. 52-59.

Lientz, B.P. and Swanson, E.B., "Impact of Development Productivity Aids on Application System Maintenance," Data Base, V. 11, N. 3 (1980), p. 114-120.

Lyngbaek, P. and McLeod, D., "Object Management in Distributed Information Systems," ACM Transactions on Office Information Systems, V. 2, N. 2 (April 1984), p. 96-122.

Microsoft Corp., Disk Operating System, IBM Corp. document #1502343, Boca Raton, Fl, Sept. 1983.

Montgomery, D. and Urban, G., Management Science in Marketing, Prentice-Hall, Englewood Cliffs, NJ, 1969.

Sirbu, M., "Innovation strategies in the electronic mail marketplace," Telecommunications Policy, Sept. 1978, p. 191-210.

Sirbu, M., and Zwimpfer, L., "Standards Setting for Computer Communications: The Case of X.25," CISR Working Paper, N. 117, Sept. 1984.

Winston, P.H., Artificial Intelligence, Addison-Wesley, Reading, MA, 1984, 527 pp.

Wurts, J.S., "The Future of Financial Modeling Systems," Proceedings of a Conference on the Future of Corporate Planning and Modeling Software Systems, Duke University, Durham, NC, June 25-26, 1981.

Zdonik, S., Object Management System Supporting Concepts: Integrated Office Workstation Applications, unpublished Ph.D. dissertation, M.I.T., May, 1983.